

### **I. DATOS ADMINISTRATIVOS**

<b>CURSO</b>	PRUEBAS DE SOFTWARE
<b>CÓDIGO</b>	1926
<b>CICLO</b>	Sexto
<b>SEMESTRE</b>	2017 – II
<b>HORAS</b>	3LA + 1VT
<b>CARRERA</b>	Computación e Informática
<b>REQUISITOS</b>	Gestión de Proyectos de TI (0340)

### **II. INTRODUCCIÓN**

Pruebas de Software es un curso que pertenece a la línea de carrera y se dicta en la carrera profesional de Computación e Informática. Brinda los conceptos básicos relacionados al área de control de la calidad de software y administración de pruebas de software, alineados a las mejores prácticas en desarrollo de software.

El curso es teórico práctico: consiste en sesiones teóricas acompañadas con aplicaciones prácticas. En primer lugar, se explica la importancia de la verificación y validación de software para el control de calidad del producto de software y se presenta una introducción al Test Driven Development. Continúa con la presentación de los fundamentos del Rational Functional Tester para la creación de scripts de pruebas funcionales. Por último, se concluye con la aplicación del Rational Performance Tester para el diseño, creación, ejecución y análisis de pruebas de rendimiento.

### **III. LOGRO DEL CURSO**

Al término del curso, el alumno desarrolla y expone los artefactos de prueba de una aplicación desarrollada bajo las plataformas Java SE y Java EE y/o aplicaciones web.

### **IV. METODOLOGÍA**

El curso aplica la metodología de taller integrador. En ese sentido, recurre a técnicas de la metodología activa, el estudio significativo y el trabajo cooperativo. Por esta razón, las actividades que se realizan en clases exigen la participación activa del alumno y la constante práctica de las herramientas y de las técnicas presentadas en laboratorio. Inmediatamente después de la presentación de cada tema, el alumno debe transferir lo aprendido mediante actividades dirigidas y dinámicas grupales. Posteriormente los alumnos deben plasmar lo ejercitado en su proyecto.

### **V. MEDIOS Y MATERIALES**

En el desarrollo del curso, se utilizan los siguientes medios y materiales:

<b>Equipamiento</b>	<b>Material Educativo y Recursos Digitales</b>
<ul style="list-style-type: none"> <li>• Computadora personal (docente)</li> <li>• Computadora personal (alumnos)</li> <li>• Proyector multimedia</li> <li>• Pizarra</li> </ul>	<ul style="list-style-type: none"> <li>• Diapositivas</li> <li>• Manual</li> <li>• IBM Rational Functional Tester</li> <li>• IBM Rational Software Performance Tester</li> </ul>

## VII. UNIDADES DE APRENDIZAJE

UNIDAD 1. Fundamentos de Pruebas de software		Duración: 12 horas
<b>Logro de la Unidad de Aprendizaje</b> Al término de la unidad, el alumno reconoce la importancia de las Pruebas de Software para el control de calidad del producto de software. Asimismo asimila los conceptos y aplica Test Driven Development (TDD) en una aplicación Java		
Capacidades	Conocimientos	
1. Identifica los distintos tipos de prueba dentro de la terminología del Test Driven Development (TDD). 2. Crea y ejecuta pruebas unitarias con JUnit.  <b>Avance 1 de proyecto final: Pruebas Unitarias con TDD</b>	<b>1.1 Tema 1: Introducción a TDD (3 hrs)</b> 1.1.1. TDD y Tipos de Pruebas 1.1.1.1. Pruebas de Aceptación 1.1.1.2. Pruebas Funcionales 1.1.1.3. Pruebas de Sistema 1.1.1.4. Pruebas Unitarias 1.1.1.5. Pruebas de Integración 1.1.2. Pruebas Unitarias  <b>1.2. Tema 2: Pruebas Unitarias con el framework JUnit (7 horas)</b> 1.2.1. Creación y Ejecución de una prueba unitaria simple con JUnit en Netbeans o Eclipse (2hrs) 1.2.2. Creación y Ejecución de una prueba unitaria de regular complejidad con JUnit en Netbeans o Eclipse (2hrs) 1.2.3. Creación y Ejecución de una prueba unitaria avanzada con JUnit en Netbeans o Eclipse (3hrs)  <b>Evaluación de Laboratorio 1 (2 hrs)</b>	

UNIDAD 2. Fundamentos Rational Functional Tester		Duración: 16 horas
<b>Logro de la Unidad de Aprendizaje</b> Al término de la unidad, el alumno crea y agrega características avanzadas a los scripts de pruebas funcionales en una aplicación Java.		
Capacidades	Conocimientos	
1. Configura aplicaciones en RFT para pruebas funcionales automatizadas. 2. Graba y reproduce scripts en RFT. 3. Analiza resultados de las pruebas realizadas en RFT.  <b>Avance 2 de proyecto final: Scripts de pruebas funcionales</b>  <b>Avance 3 de proyecto final: Scripts de pruebas funcionales con características avanzadas</b>	<b>2.1 Tema 3: Introducción al Rational Functional Tester (3 hrs)</b> 2.1.1 Arquitectura de Rational Functional Tester 2.1.2 Configuración del entorno de pruebas 2.1.3 Configuración de aplicaciones Java a probar  <b>2.2 Tema 4: Script de pruebas funcionales (9 hrs)</b> 2.2.1 Grabación de un script 2.2.2 Reproducción de un script  <b>Evaluación de Laboratorio 2 (2 hrs)</b>  2.2.3 Revisión de los resultados 2.2.4 Características avanzadas de script de pruebas  <b>Evaluación de Laboratorio 3 (2 hrs)</b>	

UNIDAD 3. Fundamentos Rational Performance Tester		Duración: 17 horas
<b>Logro de la Unidad de Aprendizaje</b> Al término de la unidad, el alumno realiza pruebas de rendimiento para su proyecto final, el cual permite resolver los desafíos de pruebas de rendimiento más comunes.		
Capacidades	Conocimientos	
1. Graba y reproduce scripts en RPT. 2. Analiza resultados de las pruebas realizadas en RPT.  <b>Avance 4 de proyecto final: Pruebas de rendimiento</b>	<b>3.2 Tema 5: Introducción al Rational Performance Tester (3 hrs)</b> 3.2.1 Arquitectura de Rational Performance Tester 3.2.2 Características y beneficios de Rational Performance Tester  <b>3.3 Tema 6: Pruebas de rendimiento (8 hrs)</b> 3.3.1 Crear y ejecutar pruebas de rendimiento 3.3.2 Análisis de resultados 3.3.3 Modificar pruebas de rendimiento  <b>Evaluación de Laboratorio 4 (2 hrs)</b>  <b>Examen Final de Laboratorio (2 hrs)</b>  <b>Sustentación de Proyecto (2hrs)</b>	

## VI. EVALUACIÓN

$$PF = 20\% \text{ PROM}(\text{EL}, 2, 1) + 20\% (\text{EL}4) + 5\% (\text{NA}1) + 10\% (\text{EV}1) + 45\% (\text{LF}1)$$

**Donde.-**  
**PF** = Promedio Final  
**EL** = Evaluaciones de Laboratorio  
**LF** = Examen Final de Laboratorio  
**EV** = Evaluación Virtual  
**NA** = Evaluación Actitudinal

EVALUACIÓN	LOGRO A EVALUAR
Evaluaciones de Laboratorio	Indicado en las unidades de aprendizaje respectivas
Examen Final de Laboratorio	TDD, elaboración de scripts de pruebas funcionales en RFT y desarrollo de pruebas de rendimiento en RPT.
Proyecto	Desarrollo de un Proyecto que somete un Software a las Pruebas respectivas y que incluye TDD y su configuración en RFT y RPT.

EVALUACIÓN	SEMANAS
------------	---------

	LABORATORIO
Evaluación 1	04
Evaluación 2	07
Evaluación 3	10
Evaluación 4	13
Examen Final	14
Nota Actitudinal	15

**Consideraciones.-**

- La nota mínima aprobatoria es 13.
- Se elimina la menor de las tres primeras Evaluaciones de Laboratorio.
- La cuarta Evaluación de Laboratorio no se elimina.

## VII. BIBLIOGRAFÍA DEL CURSO

### Básica

CARLOS BLÉ JURADO Y COLABORADORES  
2010 Diseño Ágil con TDD.  
www.iExpertos.com

DAVIS CHIP, DANIEL CHIRILLO, DANIEL GOUVEIA, FARIZ SARACEVIC, JEFFREY BOCARSLY, LARRY QUESADA, LEE THOMAS, and MARC VAN LINT  
2009 Software Test Engineering with IBM Rational Functional tester. The Definitive Resource.  
Upper Saddle River, NJ : IBM Press  
<http://www.books24x7.com/marc.asp?bookid=32653>

IBM COURSES  
2009 Essentials of IBM Rational Performance Tester, V8.0.

MARIO PIATTINI, FELIX GARCÍA, ISMAEL CABALLERO  
2007 Calidad de Sistemas Informáticos.  
México, D.F.: Alfaomega.  
(005.14 PIAT/C)

### Complementaria

IBM DEVELOPER WORKS  
2005 RUP para ingenieros de pruebas. Consultado del siguiente enlace  
<http://www.ibm.com/developerworks/rational/library/04/r-3239/index.html>